Singular Value Decomposition

and

Principal Component Analysis

What’s the deal?

Dr. Gordon McDonald
Centre for Translational Data Science,
The University of Sydney

SVD - what is it?

I have a matrix X: (centred)

$$
\begin{array}{cc}
 & \overbrace{\hspace{6cm}}^{\textbf{\textit{p} parameters, pixels, dimensions}} \\
\textbf{\textit{n} pieces of data} \left\{ \vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}} \right. &
\begin{bmatrix}
x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\
x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np}
\end{bmatrix}
\end{array}
$$

I want to write it as $X = U\Sigma V^T$

SVD - what is it?

I have a matrix X: (centred)

$$
\begin{array}{cc}
 & \overbrace{\hphantom{x_{11} \quad x_{12} \quad x_{13} \quad \cdots \quad x_{1p}}}^{\textbf{\textit{p} parameters, pixels, dimensions}} \\
\textbf{\textit{n} pieces of data} \left\{ \vphantom{\begin{matrix} x \\ x \\ \vdots \\ x \end{matrix}} \right. &
\begin{bmatrix}
x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\
x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np}
\end{bmatrix}
\end{array}
$$

X is centred = each row of X has had the average row of X subtracted from it

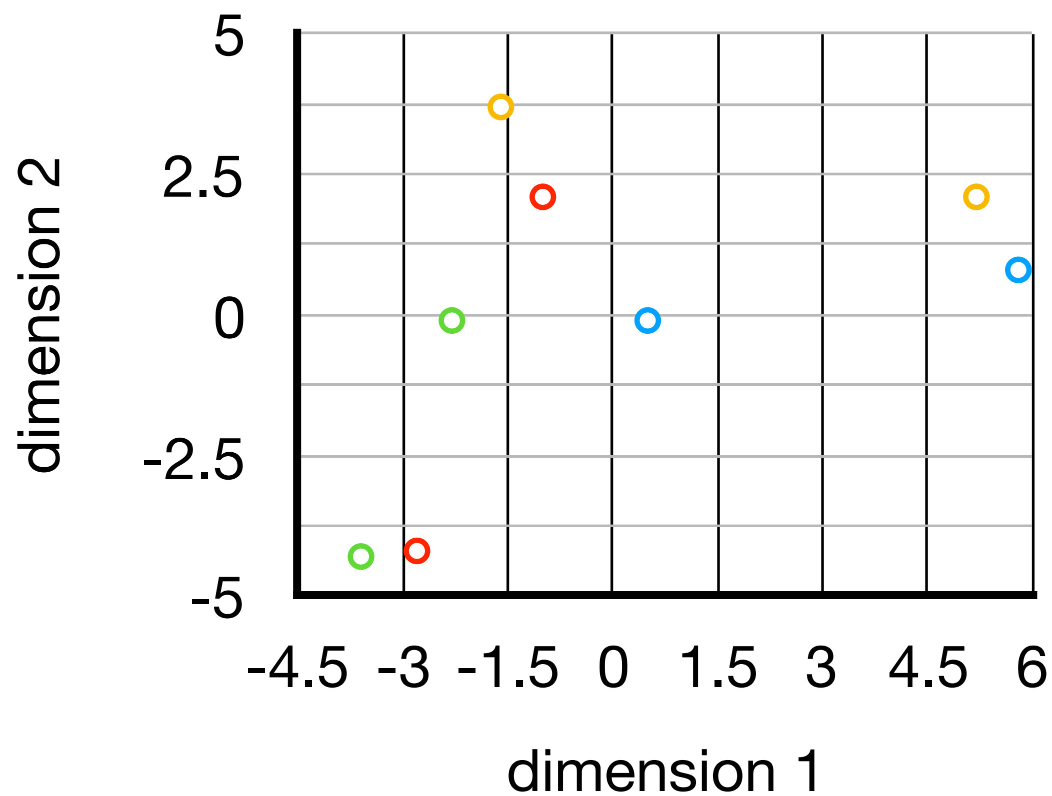
I want to write it as $X = U\Sigma V^T$

SVD - what is it?

I have a matrix X: (centred)

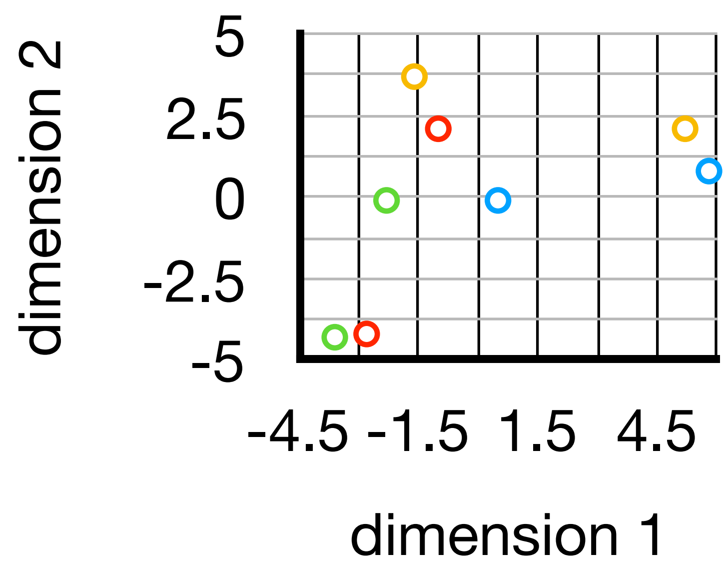
p = 2 dimensions

n = 8
pieces of data



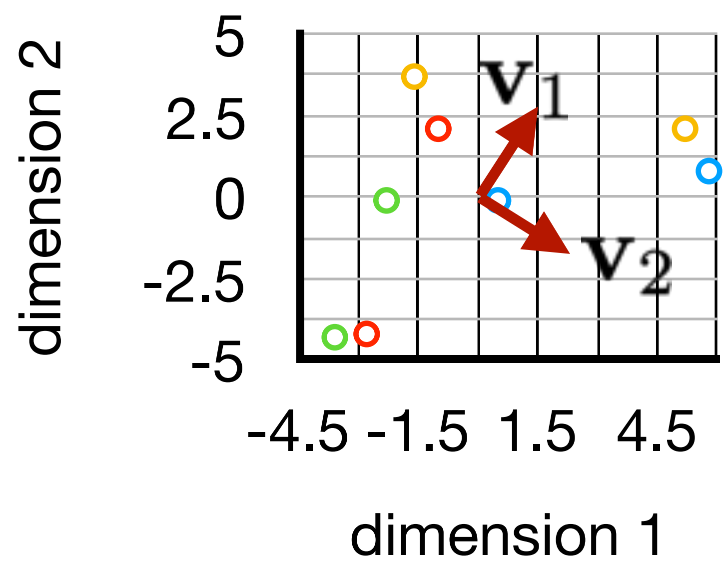
I want to write it as $X = U\Sigma V^T$

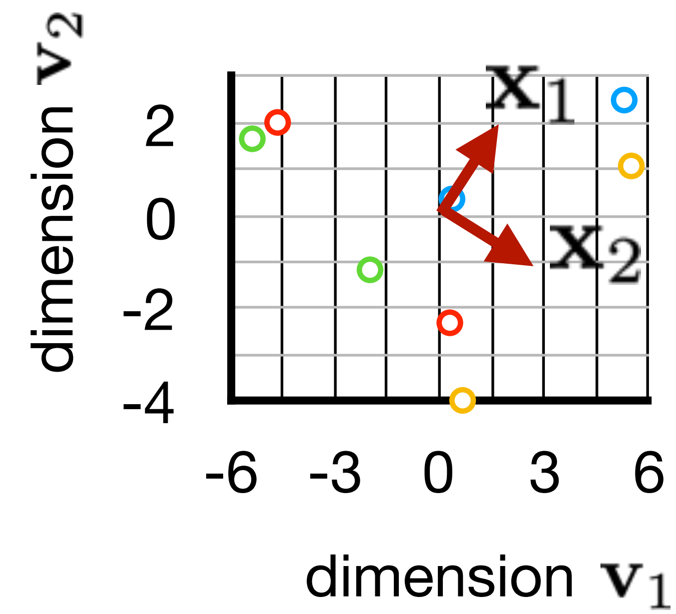
X



$$X = \overbrace{U}^{\boldsymbol{n} \textbf{ by } \boldsymbol{n}} \quad \overbrace{\Sigma}^{\boldsymbol{n} \textbf{ by } \boldsymbol{p}} \quad \overbrace{V^T}^{\boldsymbol{p} \textbf{ by p}}$$

X

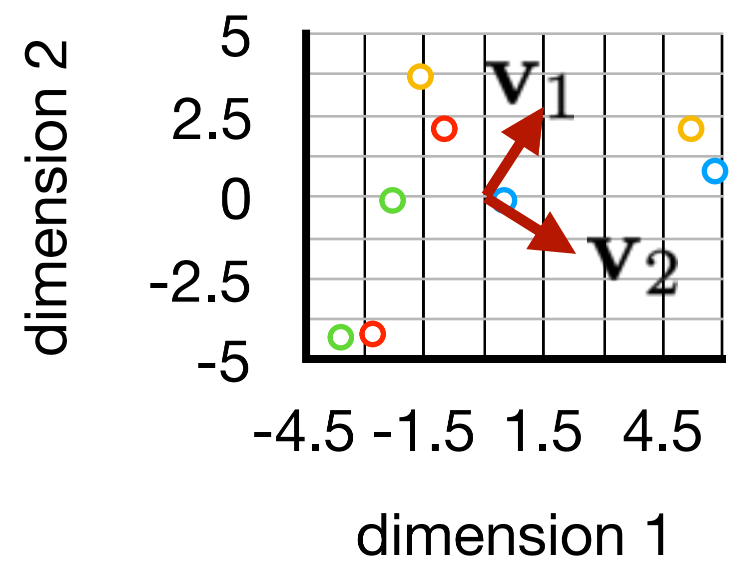


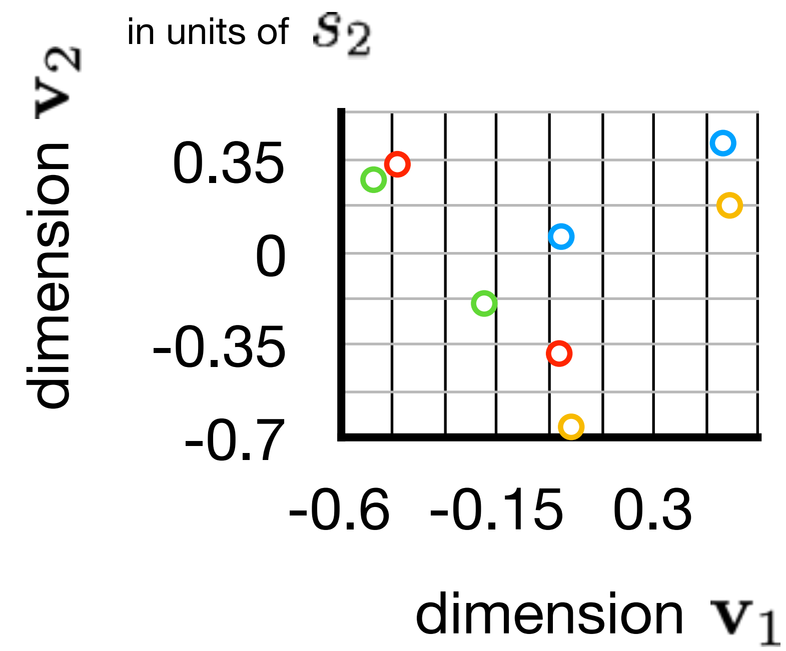


$$XV = U\Sigma$$

$$X = \overbrace{\qquad U \qquad}^{\boldsymbol{n} \textbf{ by } \boldsymbol{n}} \overbrace{\qquad \Sigma \qquad}^{\boldsymbol{n} \textbf{ by } \boldsymbol{p}} \overbrace{\begin{bmatrix} \cdots & \mathbf{v}_1 & \cdots \\ \cdots & \mathbf{v}_2 & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_p & \cdots \end{bmatrix}}^{\boldsymbol{p} \textbf{ by p}}$$

X





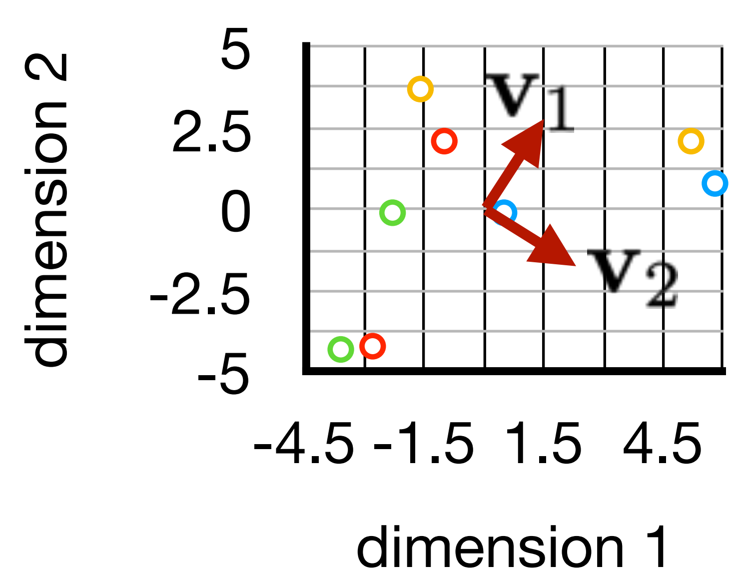
in units of s_1

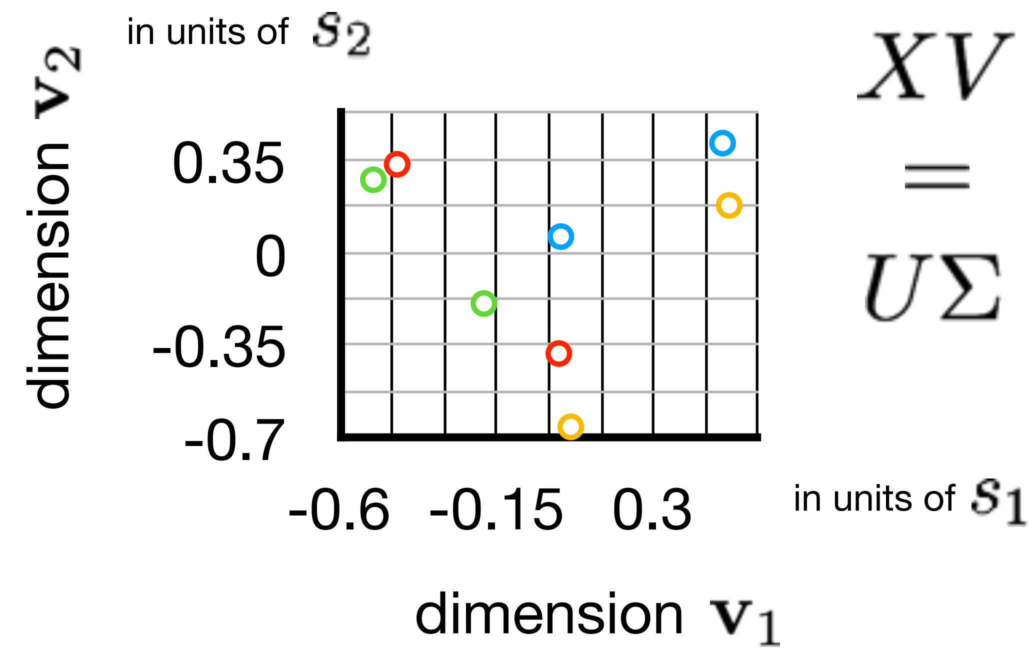
$$XV = U\Sigma$$

$$X = \underbrace{\begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}}_{\textit{n} \text{ by } \textit{n}} \underbrace{\begin{bmatrix} s_1 & 0 & 0 & \cdots & 0 \\ 0 & s_2 & 0 & \cdots & 0 \\ 0 & 0 & s_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & s_p \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\textit{n} \text{ by } \textit{p}} \underbrace{\begin{bmatrix} \cdots & \mathbf{v}_1 & \cdots \\ \cdots & \mathbf{v}_2 & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_p & \cdots \end{bmatrix}}_{\textit{p} \text{ by p}}$$

$\text{if } n > p$

X



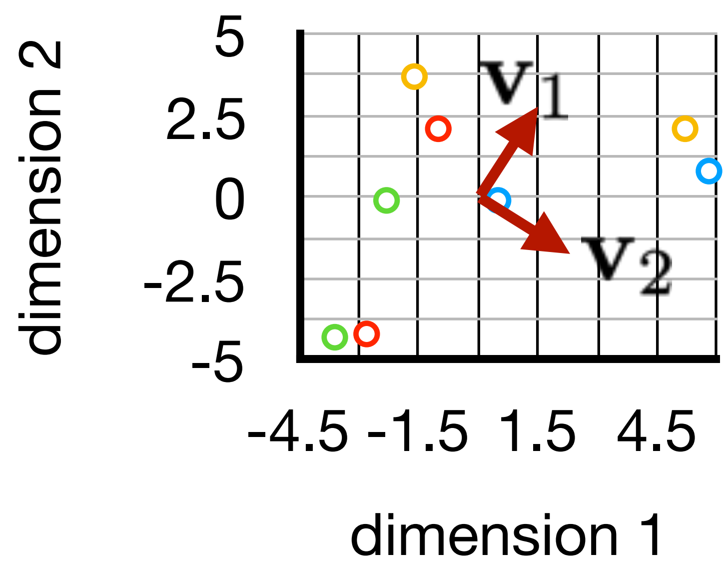


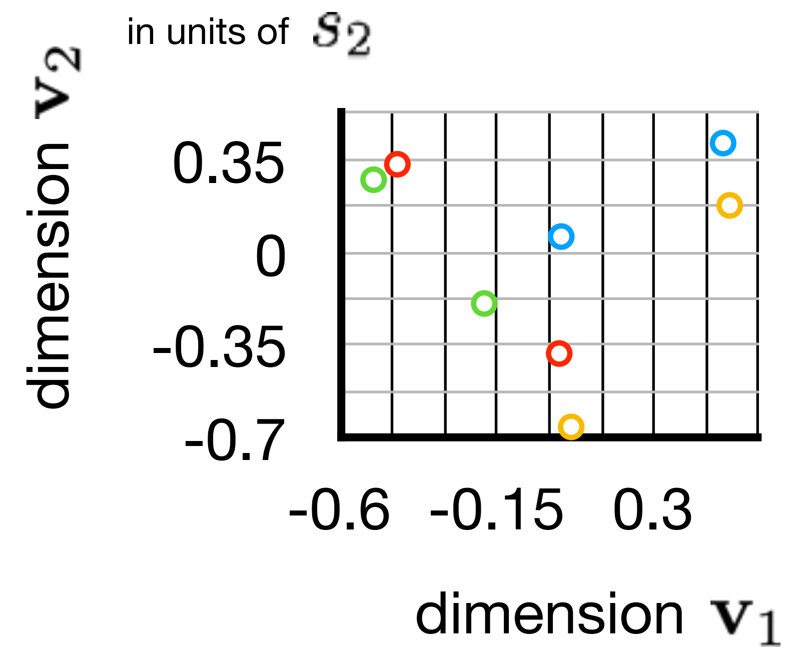
$$XV = U\Sigma$$

$$X = \overbrace{\begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}}^{\boldsymbol{n} \textbf{ by } \boldsymbol{n}} \overbrace{\begin{bmatrix} s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ 0 & 0 & s_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & s_p \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}}^{\boldsymbol{n} \textbf{ by } \boldsymbol{p}} \overbrace{\begin{bmatrix} \dots & \mathbf{v}_1 & \dots \\ \dots & \mathbf{v}_2 & \dots \\ & \vdots & \\ \dots & \mathbf{v}_p & \dots \end{bmatrix}}^{\boldsymbol{p} \textbf{ by p}}$$

$$\text{if } n > p$$

X





$XV = U\Sigma$

$$X = \overbrace{\begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ \vdots & \vdots & & \vdots \end{bmatrix}}^{n \textbf{ by } n} \overbrace{\begin{bmatrix} s_1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & s_2 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & s_3 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & s_n & 0 & \cdots & 0 \end{bmatrix}}^{n \textbf{ by } p} \overbrace{\begin{bmatrix} \cdots & \mathbf{v}_1 & \cdots \\ \cdots & \mathbf{v}_2 & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_p & \cdots \end{bmatrix}}^{p \textbf{ by } p}$$

if $p > n$

PCA - what is it?

I have a matrix X: (centered)

***p* parameters, pixels, dimensions**

***n* pieces of data**

$$
\begin{bmatrix}
x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\
x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np}
\end{bmatrix}
$$

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$

PCA - what is it?

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$

PCA - what is it?

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$

I want to find a new orthonormal coordinate system with

- most of the variance along the first axis
- each successive axis having as much as possible of the remaining variance

PCA - what is it?

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$

I want to find a new orthonormal coordinate system with

•most of the variance along the first axis

•each successive axis having as much as possible of the remaining variance

I want to find a new orthonormal coordinate system with

•most of the data explained along the first axis

•each successive axis explaining as much of the data as possible

PCA - what is it?

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$
$$= V D V^T$$

p **by p**

PCA - what is it?

I want to diagonalise its covariance matrix

$$C = \frac{1}{n-1} X^T X$$

***p* by p**

$$= V D V^T$$

$$\begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_p \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_p \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1 & \cdots \\ \cdots & \mathbf{v}_2 & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_p & \cdots \end{bmatrix}$$

***p* by p** ***p* by p** ***p* by p**

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$= \frac{1}{n-1} (V \Sigma^T U^T)(U \Sigma V^T)$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$= \frac{1}{n-1} (V \Sigma^T U^T)(U \Sigma V^T)$$

$$= \frac{1}{n-1} V \Sigma^T (U^T U) \Sigma V^T$$

This is the identity

$$U^T U = I$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$= \frac{1}{n-1} (V \Sigma^T U^T)(U \Sigma V^T)$$

$$= \frac{1}{n-1} V \Sigma^T (U^T U) \Sigma V^T$$

This is the identity

$$U^T U = I$$

$$= \frac{1}{n-1} V \Sigma^T \Sigma V^T$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$= \frac{1}{n-1} (V \Sigma^T U^T)(U \Sigma V^T)$$

$$= \frac{1}{n-1} V \Sigma^T (U^T U) \Sigma V^T$$

This is the identity

$$U^T U = I$$

$$= \frac{1}{n-1} V \Sigma^T \Sigma V^T$$

$$= \frac{1}{n-1} V \Sigma^2 V^T$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U\Sigma V^T$$

$$C = \frac{1}{n-1} (U\Sigma V^T)^T (U\Sigma V^T)$$

$$= \frac{1}{n-1} (V\Sigma^T U^T)(U\Sigma V^T)$$

$$= \frac{1}{n-1} V\Sigma^T (U^T U)\Sigma V^T$$

This is the identity

$$U^T U = I$$

$$= \frac{1}{n-1} V\Sigma^T \Sigma V^T$$

$$= \frac{1}{n-1} V\Sigma^2 V^T$$

$$= V \left(\frac{1}{n-1} \Sigma^2 \right) V^T$$

$$= VDV^T$$

$$C = \frac{1}{n-1} X^T X$$

Write X as a singular value decomposition

$$X = U \Sigma V^T$$

$$C = \frac{1}{n-1} (U \Sigma V^T)^T (U \Sigma V^T)$$

$$= \frac{1}{n-1} (V \Sigma^T U^T)(U \Sigma V^T)$$

$$= \frac{1}{n-1} V \Sigma^T (U^T U) \Sigma V^T$$

This is the identity

$$U^T U = I$$

$$= \frac{1}{n-1} V \Sigma^T \Sigma V^T$$

$$= \frac{1}{n-1} V \Sigma^2 V^T$$

$$= V \left(\frac{1}{n-1} \Sigma^2 \right) V^T$$

$$= V D V^T$$

So the diagonal matrix D is

$$D = \frac{1}{n-1} \Sigma^2$$

So we can do PCA by SVD

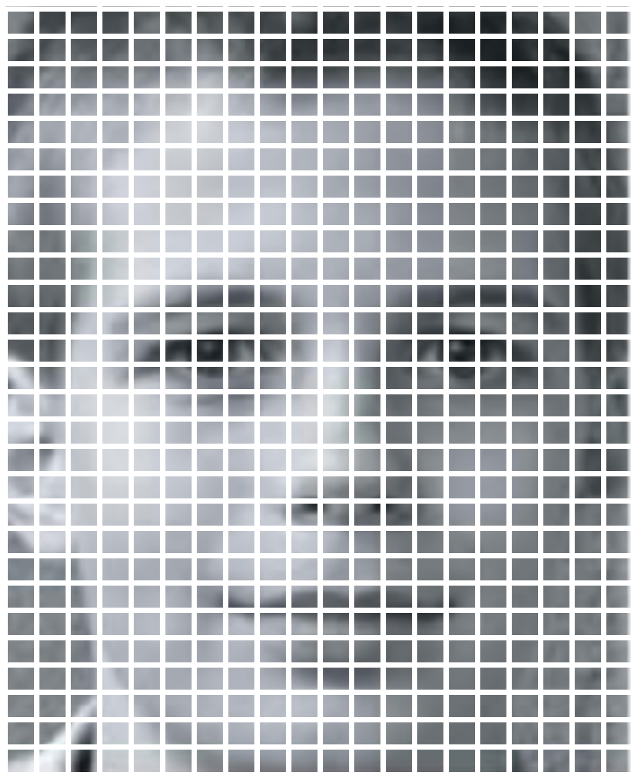
(but we could also do it by directly diagonalising the covariance matrix)

PCA using SVD is more stable against numerical rounding errors

The fastest way to do each on large data sets is probabilistically / stochastically

Example 1: image data compression n= 32 images, p~100,000 pixels

Example 1: image data compression n= 32 images, p~100,000 pixels



vectorise the image

this becomes row n=1 of X

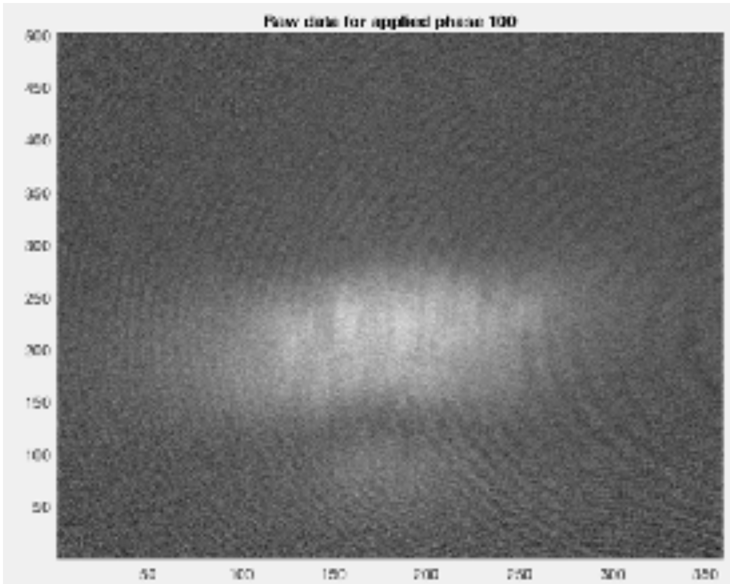
Example 1: image data compression **actual photos**

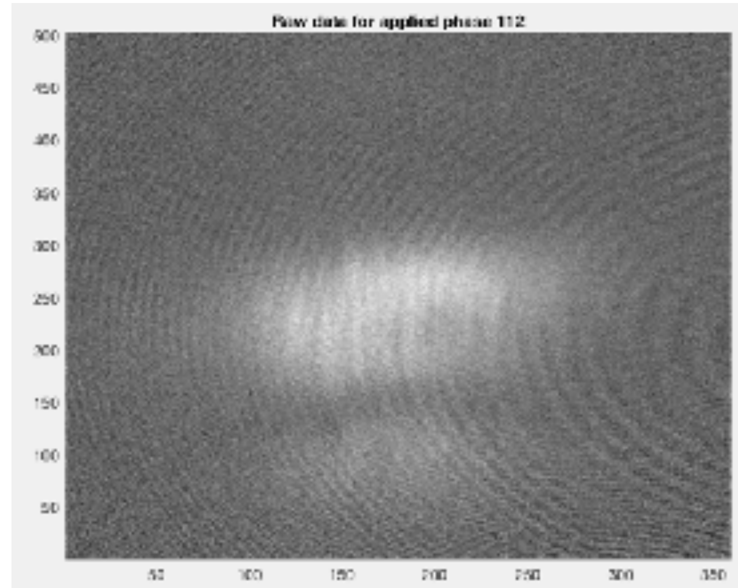
http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/11ImageProc/15PCA.pdf

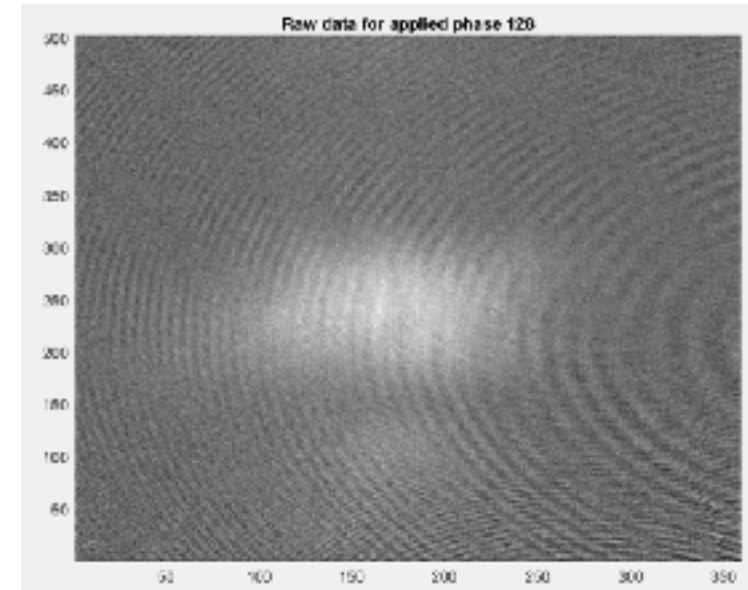
Example 1: image data compression   images reconstructed using only first 4 PCs

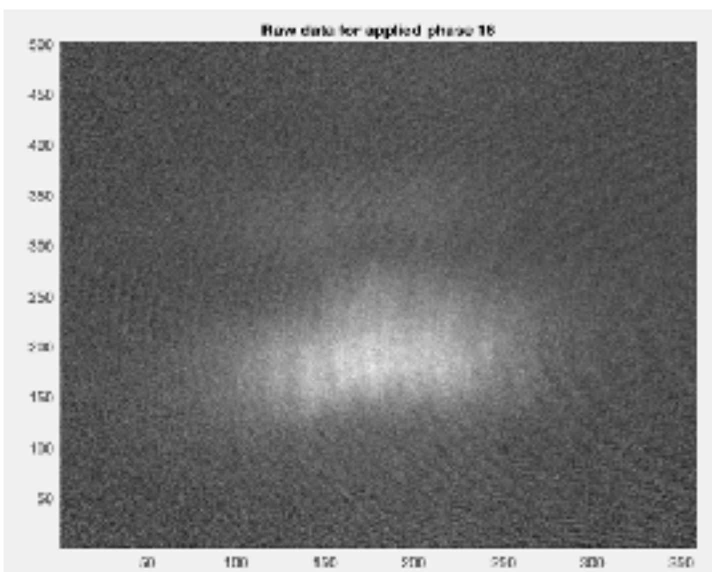
http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/11ImageProc/15PCA.pdf

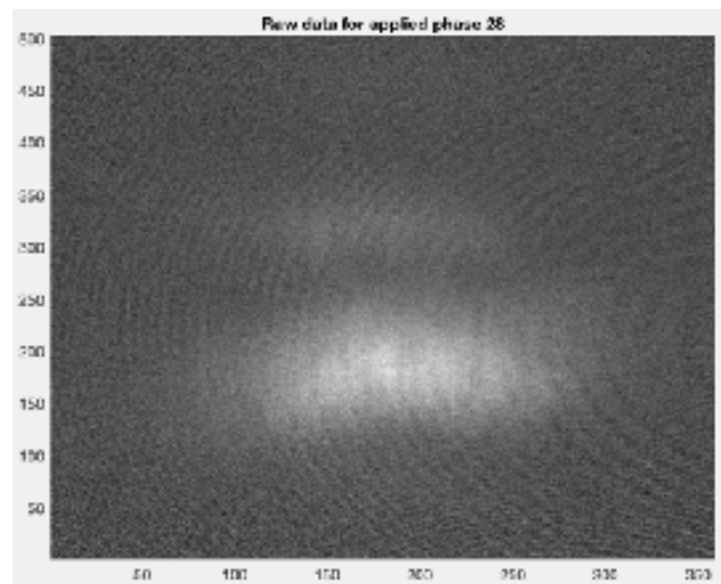
Example 2: images, data extraction n= 91 images, p~150,000 pixels

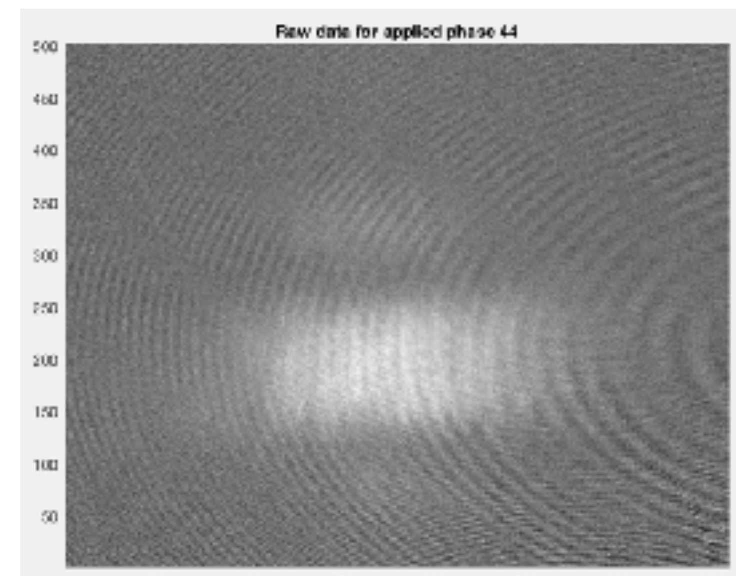


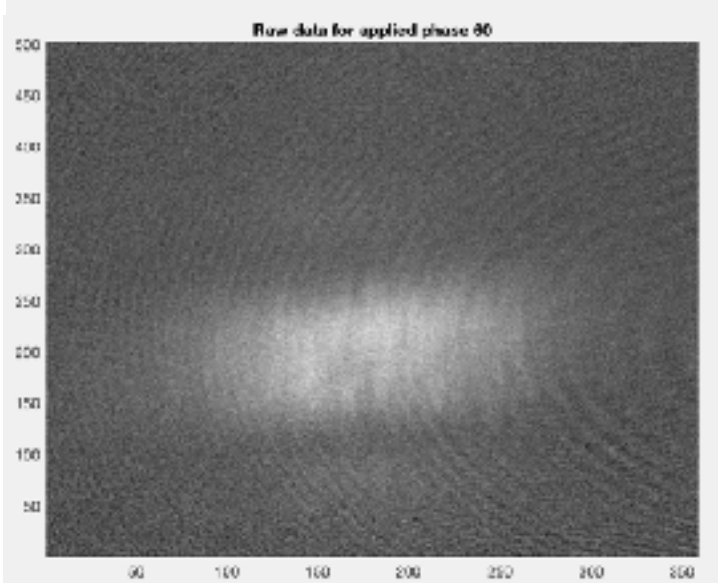


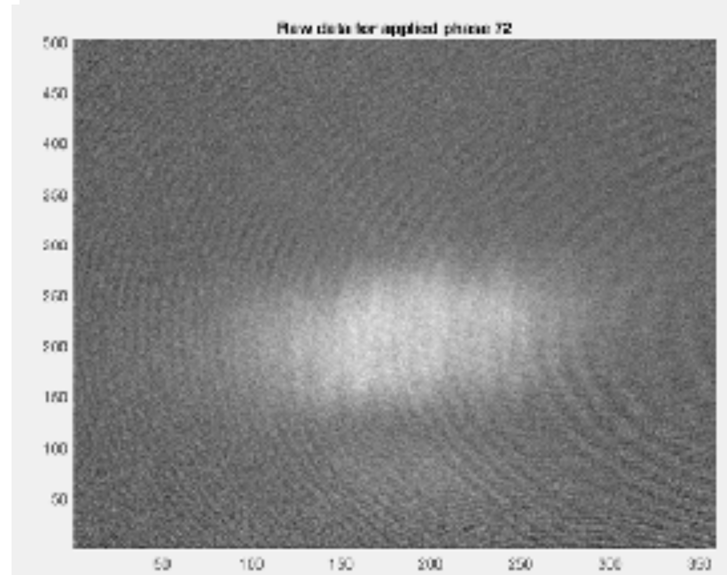


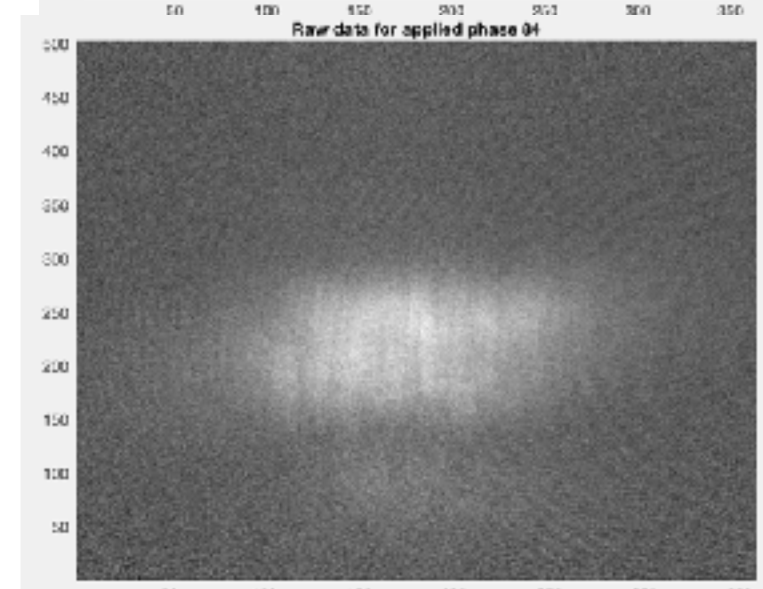




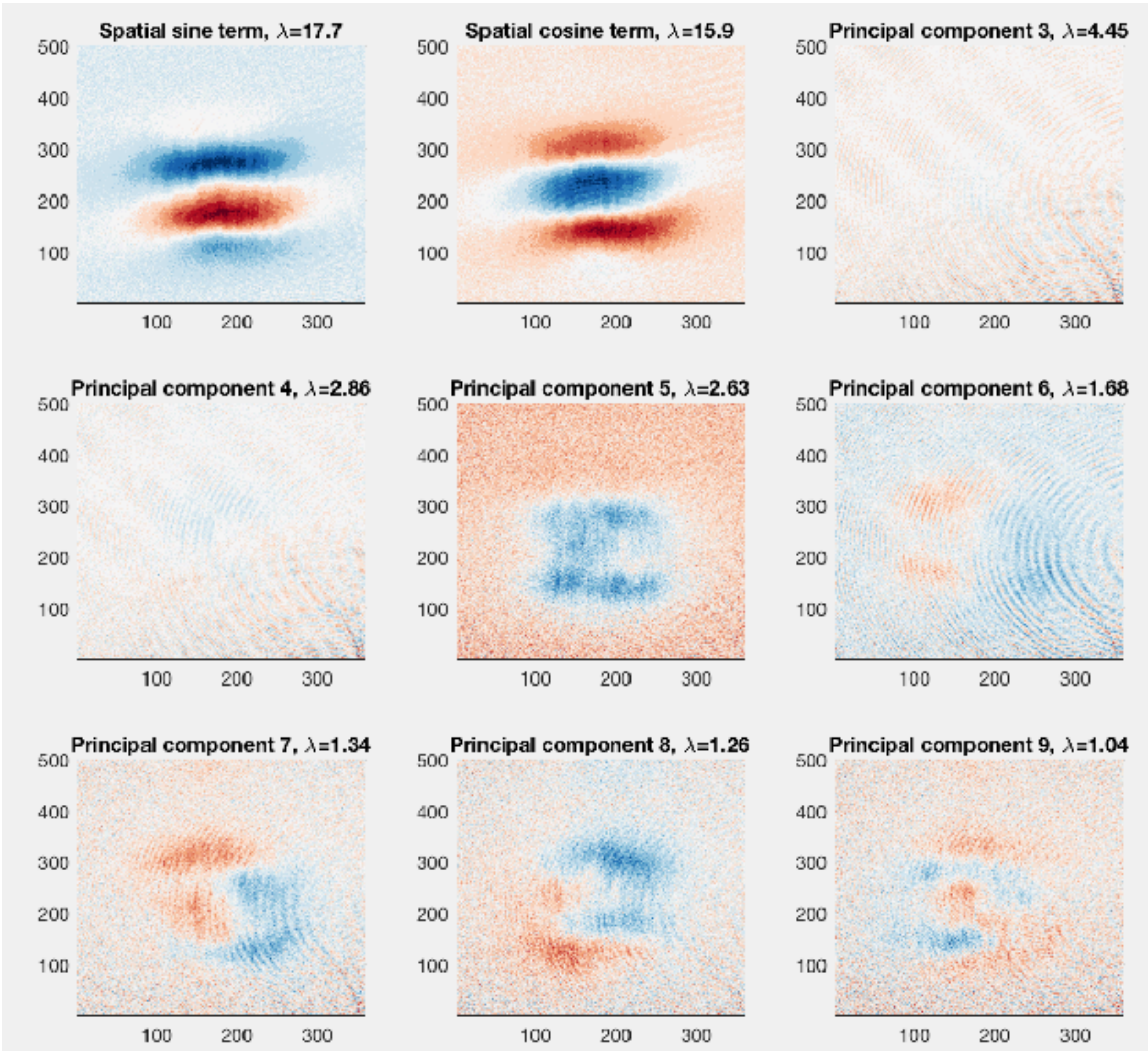




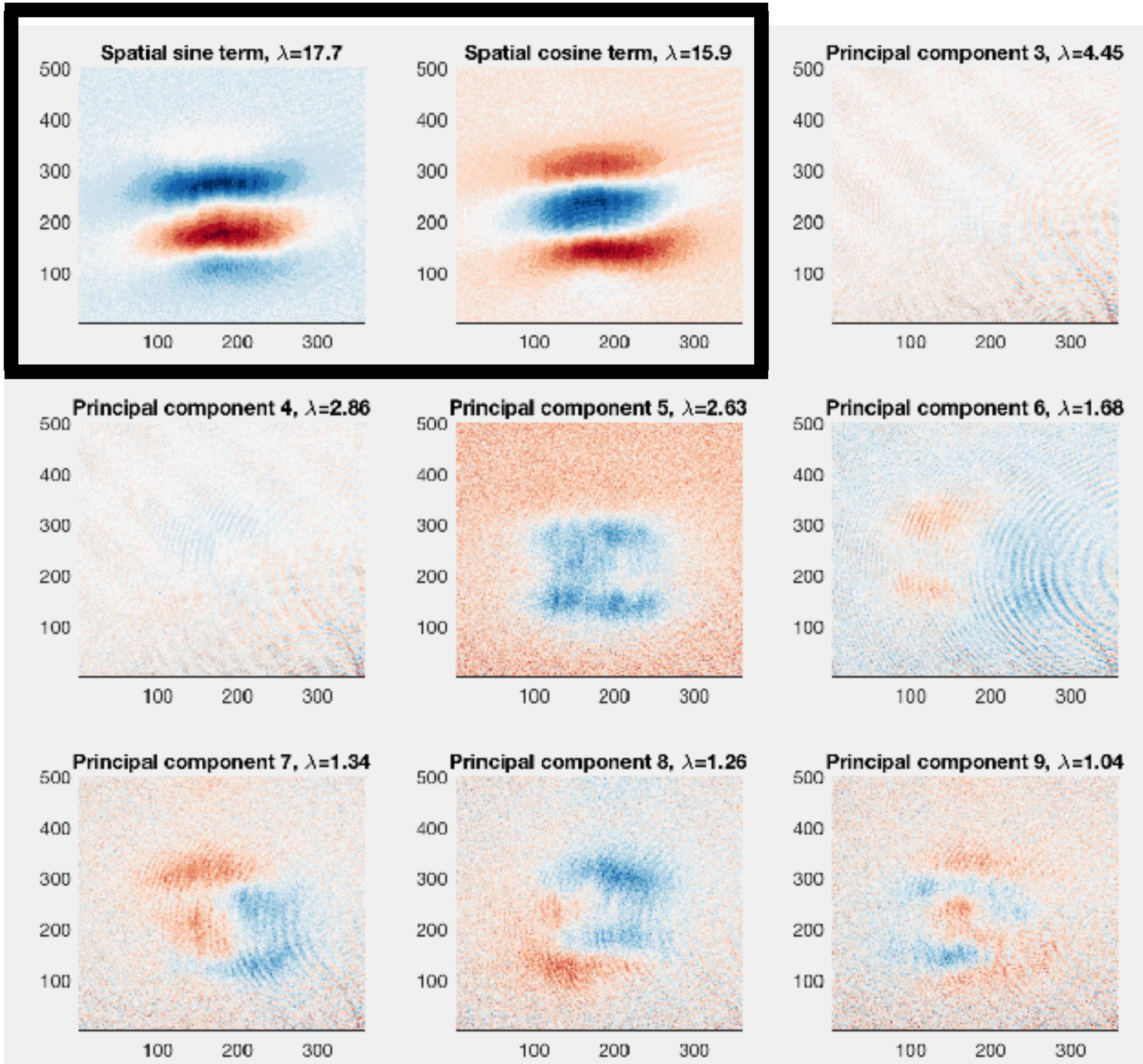




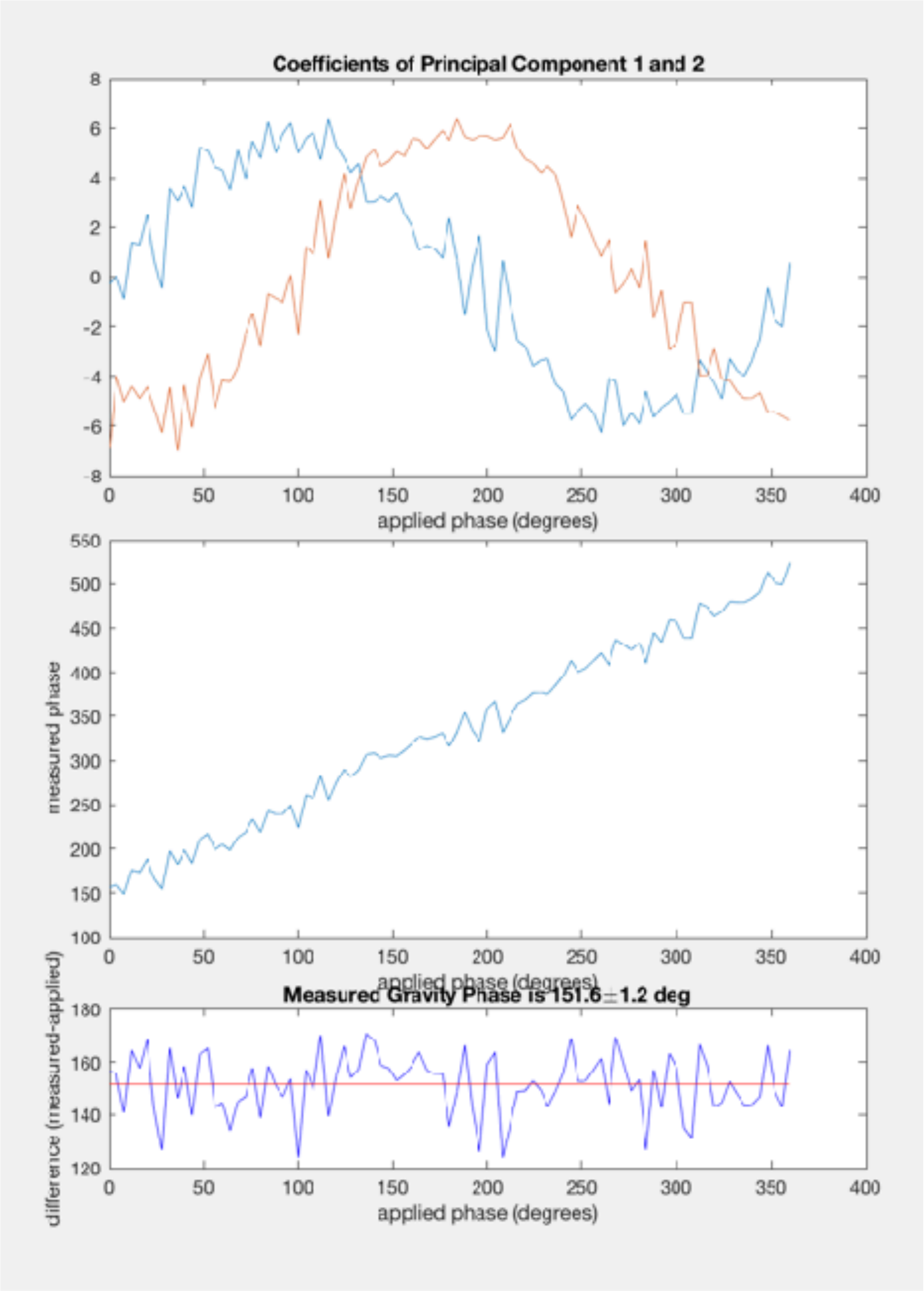
Example 2: images, data extraction **n= 91 images, p~150,000 pixels**



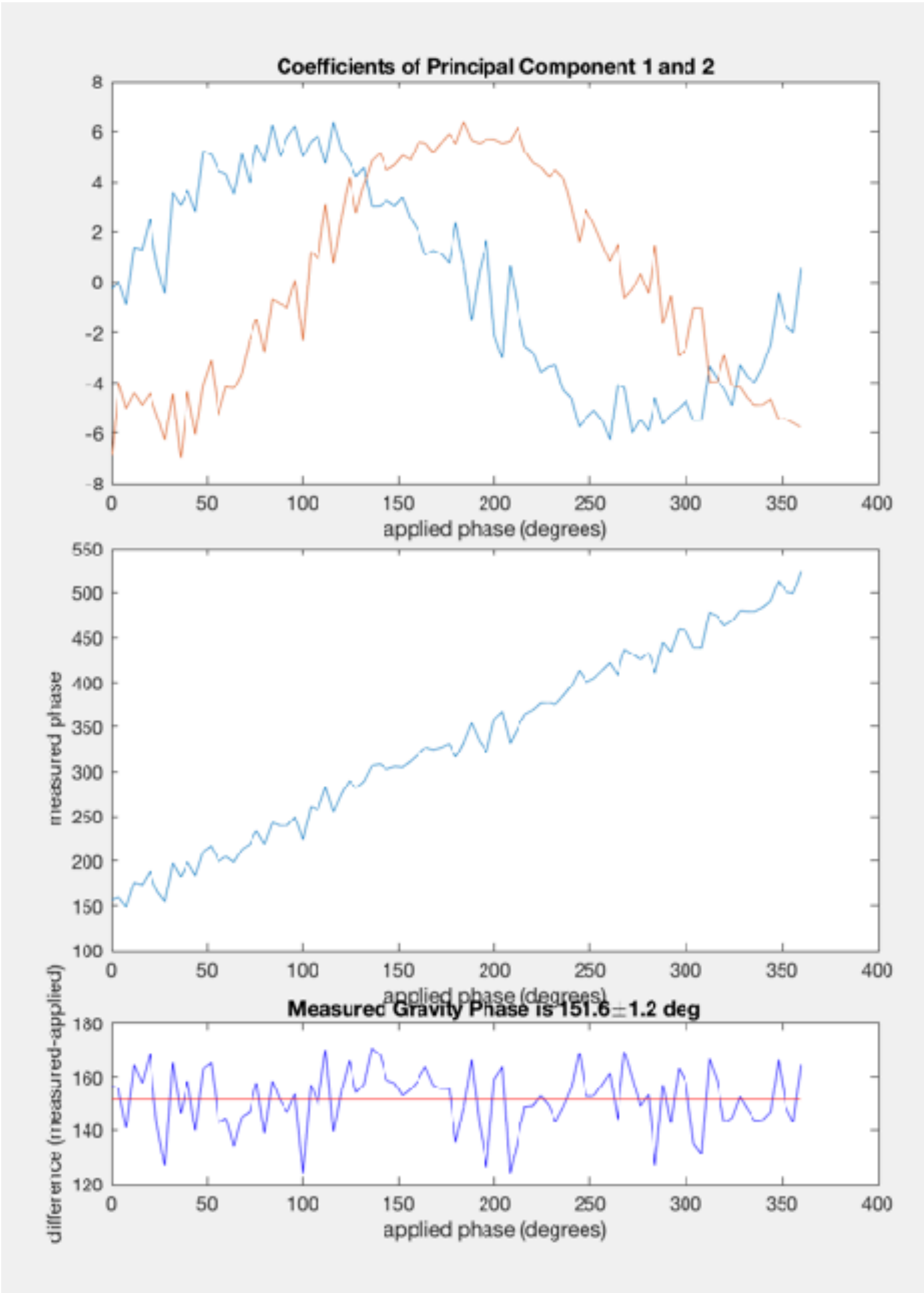
Example 2: images, data extraction **n= 91 images, p~150,000 pixels**

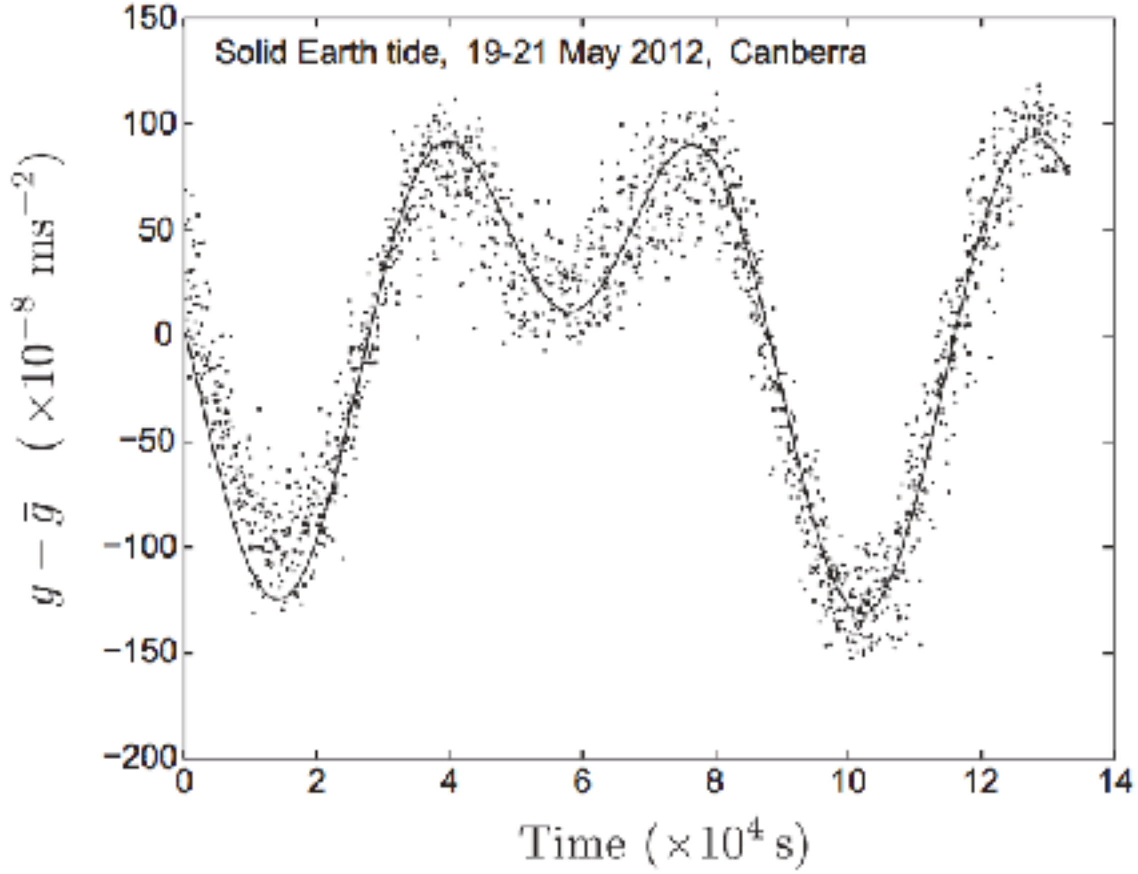


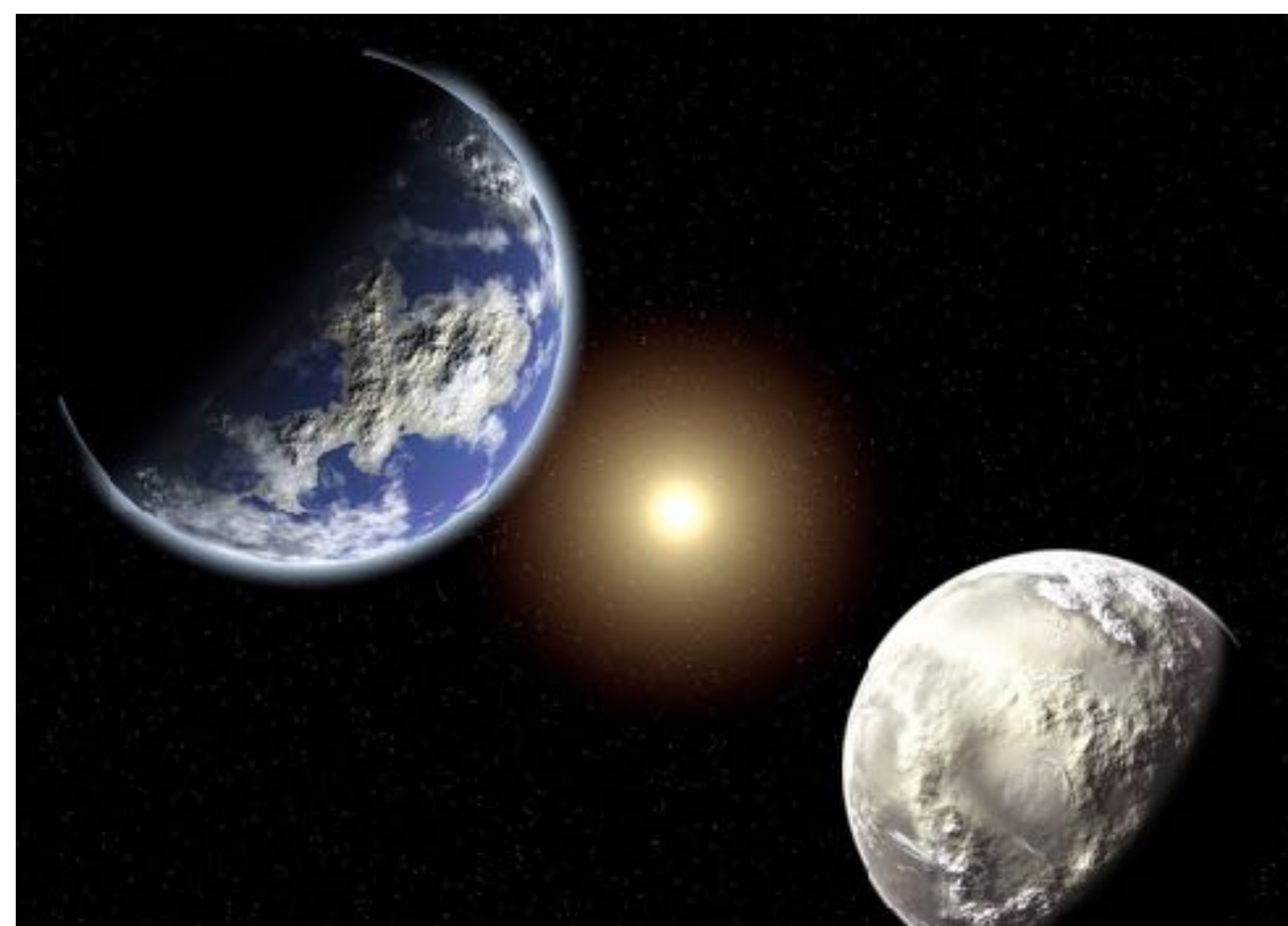
Example 2: images, data extraction   n= 91 images, p~150,000 pixels



Example 2: images   n= 91 images, p~150,000 pixels

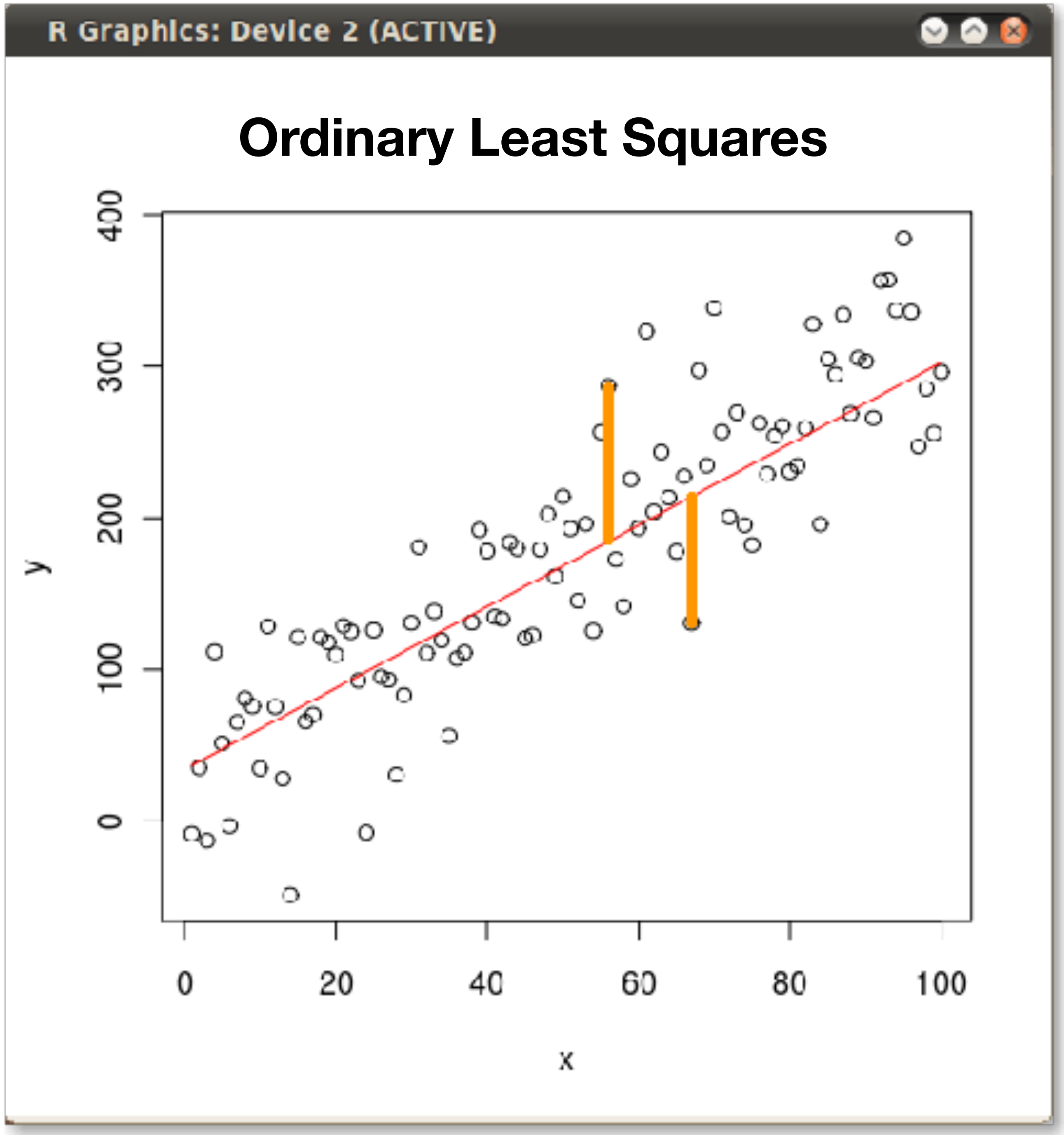






Example 3: linear fit n= 100 data points, p=2 dimensions

minimize $|y - \hat{y}|^2$ **given** $x = \hat{x}$

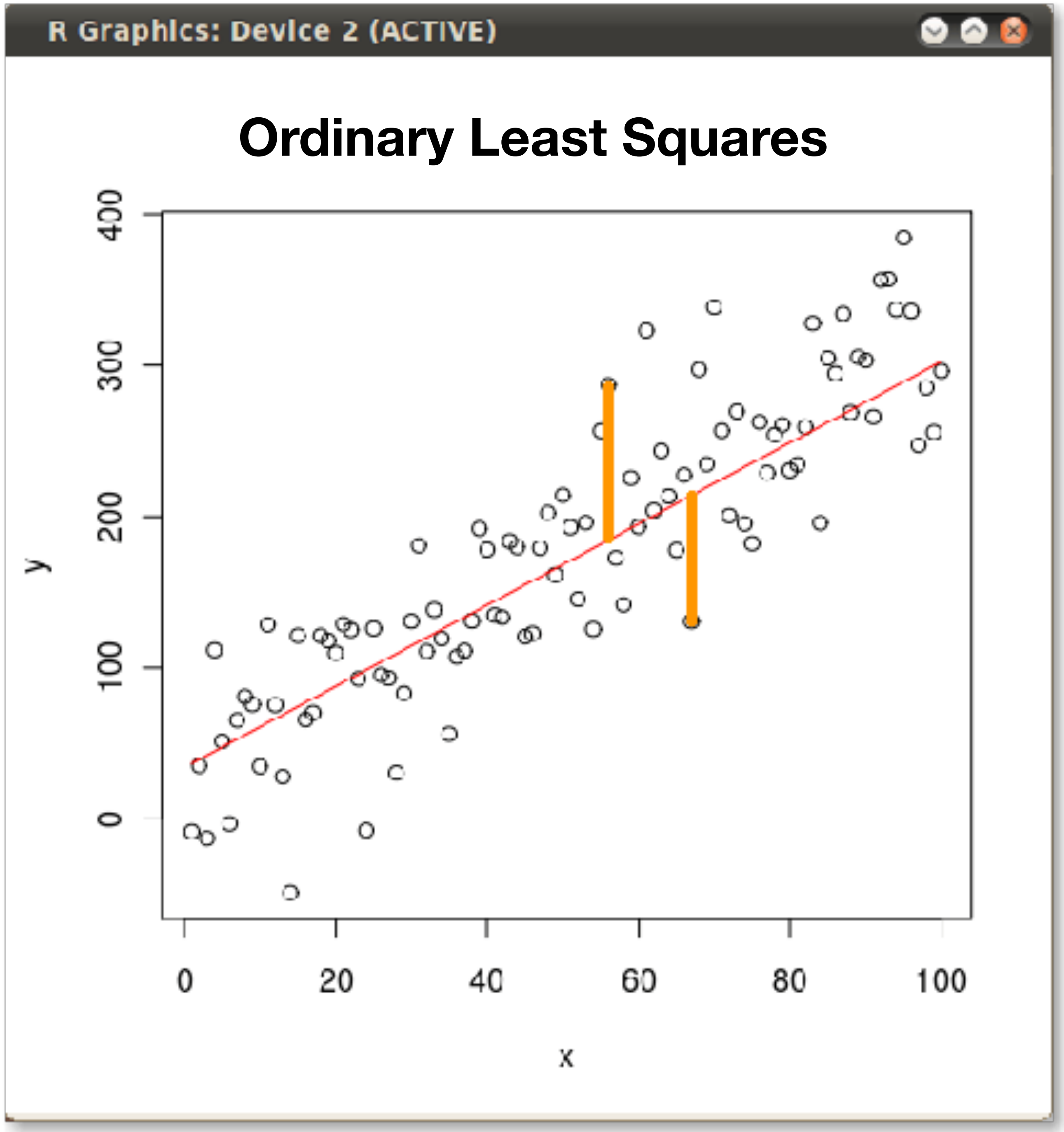


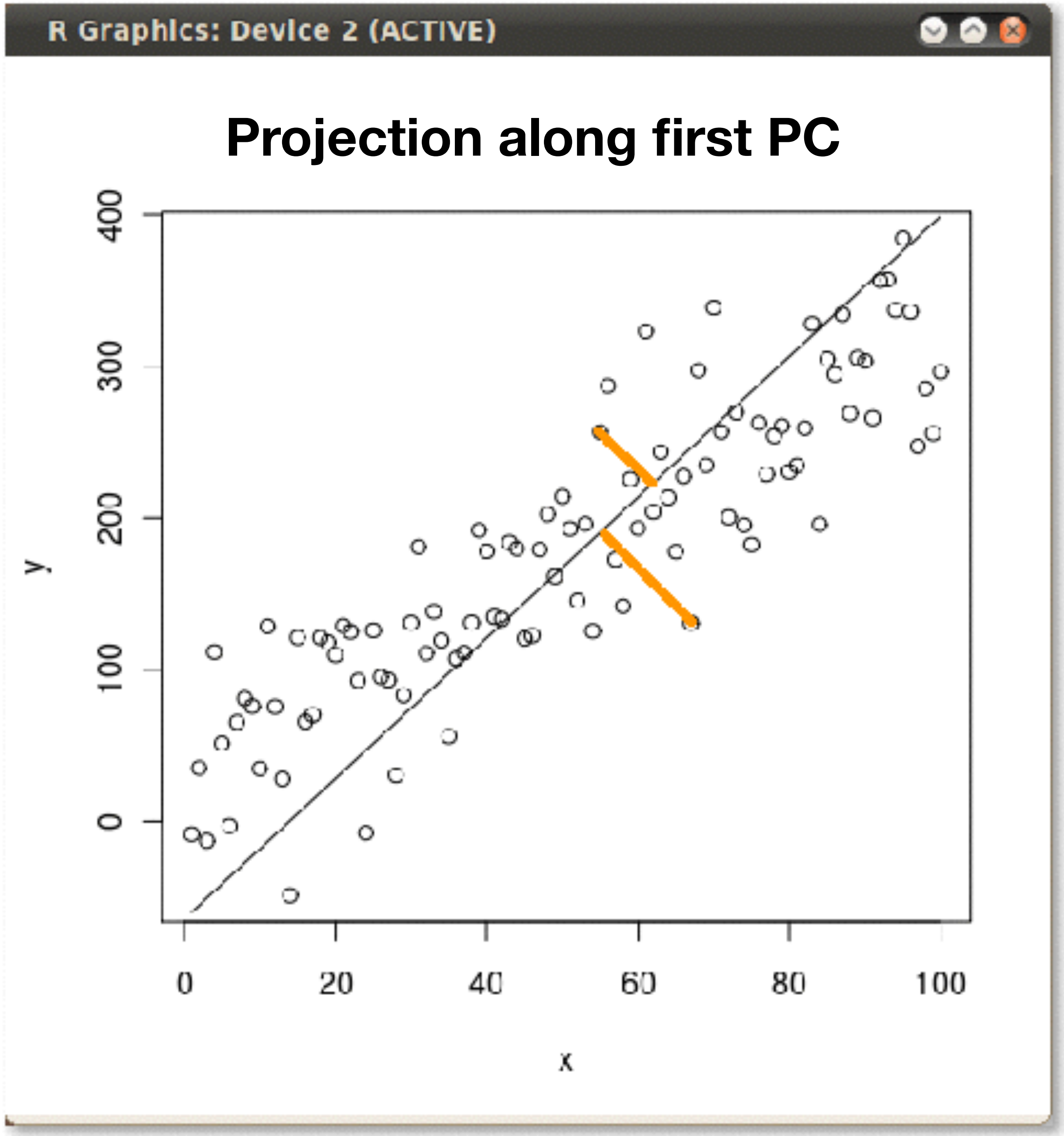
https://www.r-bloggers.com/principal-component-analysis-pca-vs-ordinary-least-squares-ols-a-visual-explanation/

Example 3: linear fit   n= 100 data points, p=2 dimensions

minimize $|y - \hat{y}|^2$ **given** $x = \hat{x}$

minimize $|y - \hat{y}|^2 + |x - \hat{x}|^2$





https://www.r-bloggers.com/principal-component-analysis-pca-vs-ordinary-least-squares-ols-a-visual-explanation/

For more authoritative references, see also:

SVD : *D. Lay,* Linear Algebra and its Aplications, Chapter 7

PCA : *C. Bishop,* Pattern Recognition and Machine Learning, Chapter 12

Or for the tl;dr…

https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca